MASTER EN SCIENCES INFORMATIQUES

Analyse quantitative de gels d'électrophorèse à une dimension par traitement d'image

Schraverus, Oscar

Award date:
1991

Awarding institution:
Universite de Namur

ANALYSE QUANTITATIVE DE GELS D'ELECTROPHORESE A UNE DIMENSION PAR TRAITEMENT D'IMAGE

SCHRAVERUS Oscar

Promoteur :

Mme M. Noirhomme-Fraiture

Année Académique 1990-1991.

Mémoire présenté pour l'obtention du diplôme de Licencié et Maître en Informatique.

Résumé

Ce travail traite de l'application des techniques de traitement d'images à l'analyse de gels d'électrophorèse à une dimension. La technique utilisée diffère des techniques classiques d'analyse de gels par traitement d'images par le fait que les algorithmes travaillent sur une coupe longitudinale des pistes (profil) permettant ainsi un gain appréciable en durée de traitement et en volume mémoire nécessaire. L'algorithme principal, qui permet la quantification des bandes se base sur une modélisation des bandes par des courbes de Gauss.

Abstract

A one dimensional gel electrophoresis analysis system using image processing techniques has been developped. The main difference with other gel analysis systems is the use of real one-dimension algorithms, acting on a single longitudinal section of each lane. The analysis can be performed very quickly, and only little memory space is used, allowing this program to run on common hardware. The main algorithm, performing the quantification of the bands uses Gauss modeling techniques.

Avant tout, je tiens à remercier Madame Noirhomme et Monsieur Remacle, promoteur et co-promoteur de ce mémoire, ainsi que toutes les personnes, qui ont contribué à la réalisation de ce mémoire par leur aide, conseils, soutient moral ou matériel...
Je pense notamment à
l'Unité de Biochimie Cellulaire,
Christine Donville et Stephane Maes,
Catherine,
Gabriel,
Marc, Marcus, Luc, Jean-Philippe....

Table des matières Page

Table des matières (suite) Page

1. INTRODUCTION

1.1 But de ce travail

L'électrophorèse à une dimension fait partie des techniques couramment utilisées en recherche biochimique. Cette technique, qui sera expliquée plus en détail au chapitre suivant, est employée de longue date. Elle permet d'obtenir rapidement une estimation du poids moléculaire de divers constituants d'un mélange de protéines, dont la composition est éventuellement inconnue à priori.

Plus récemment, a été développée la technique d'électrophorèse bidimensionnelle. Cette dernière permet une bien meilleure séparation des protéines, étant donné que l'on se base ici sur deux critères. En contrepartie, la quantité d'information à traiter est gigantesque. C'est pour cette raison qu'il est intéressant d'utiliser des techniques avancées de traitement de l'information. Le traitement d'image s'est avéré très utile pour analyser les gel d'électrophorèse à deux dimensions. Travaux et recherhes sur ce sujet ne manquent pas. Au laboratoire de Biochimie Cellulaire des Facultés Universitaires Notre Dame de la Paix à Namur, un mémoire a été consacré à ce sujet (*).

Etant donné que l'électrophorèse à une dimension est utilisée beaucoup plus souvent, l'idée d'utiliser les techniques d'analyse mises au point pour les gels bidimensionnels au profit des gels à une dimension vit le jour. Ceci dans le but d'accroître la précision de l'analyse des résultats.

(*) V.Henin 1988-1989

Le but de ce travail est donc la mise au point d'un système d'analyse de gels d'électrophorèse à une dimension utilisant les techniques de traitement d'image, se basant sur l'expérience acquise dans le traitement des gels à deux dimensions.

1.2. Connaissances et expérience disponibles

Contrairement au cas de l'analyse de gels à deux dimensions, il existe peu de travaux traitant de l'analyse de gels à une dimension.
Il aurait été possible d'adapter les procédures développées pour le traitement de gels à deux dimensions. Cependant, la spécificité du problème d'une part, et la possibilité d'un gain de temps de traitement substantiel sont les raisons pour lesquelles de nouvelles procédures ont été créées dans le cadre de ce travail.

2. LA TECHNIQUE D'ELECTROPHORESE

Dans ce chapitre sera expliquée succintement la technique d'électrophorèse. Pour bien comprendre son principe, nous parlerons un peu de la nature des protéines.

2.1. Nature des protéines

Les protéines sont des macromolécules constituées d'acides aminés. Les acides aminés, dont la formule générale est

$$H_2N\text{-}CHR\text{-}COOH$$

se lient les uns aux autres par leurs groupement amine (-NH2) et carboxyle (-COOH), formant ainsi de longues chaînes. Ce sont les groupements -R liés à l'atome de carbone central qui différencient les acides aminés les uns des autres. Ces derniers peuvent être neutres, acides ou basiques, et peuvent inclure des groupements sulfhydriles (-SH). Les ponts disulfure (-S-S-) formés à partir de deux radicaux sulfhydriles, ainsi que les forces d'attraction électrostatique entre les radicaux chargés positivement et négativement en fonction de leur caractère basique ou acide sont les principaux facteurs déterminant la conformation spatiale de la protéine. La longueur de la chaîne quant à elle détermine son poids moléculaire.

2.2. L'électrophorèse

Nous venons de voir que les protéines sont constituées d'acides aminés à charge neutre, positive ou négative. Ceci résulte en une charge globale de la protéine. Cette charge varie en fonction du pH du milieu, car un acide libère ses ions H+ en milieu neutre ou alcalin (COOH --> COO-),un radical basique au contraire les capte (-NR2 --> -NRH+).

Cette propriéte des protéines est mise à profit pour les séparer, soit dans un gradient de pH, soit à pH constant.

Dans le premier cas, les protéines vont migrer jusqu'à l'endroit où leur charge globale est neutre ; c'est le point isoélectrique.

Dans l'autre cas, elles migreront en fonction de leur poids moléculaire : elles migreront d'autant plus vite que leur taille est petite.

L'électrophorèse à deux dimensions combine ces deux techniques de séparation, tandis que l'électrophorèse à une dimension utilise souvent la séparation sur base du poids moléculaire.

En pratique, on procède comme suit :

L'échantillon à analyser est placé sur le bord d'une plaque recouverte d'un gel de polyacrylamide, disposé en pistes. Lorsque le gel est soumis à un champ électrique orienté parallèlement aux pistes, les protéines migrent dans les pistes, donnant lieu à la formation de bandes. Ces bandes sont révélées par une technique ad hoc, par exemple une coloration spécifique. Un mélange étalon permet de déduire le poids moléculaire des constituants de l'échantillon en fonction de la distance parcourue (distance de migration).

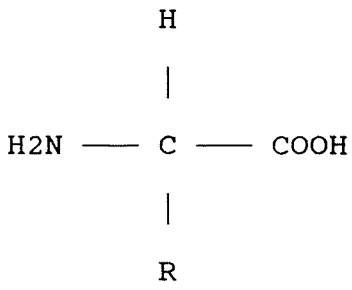


Figure 1 : Structure d'acide aminé

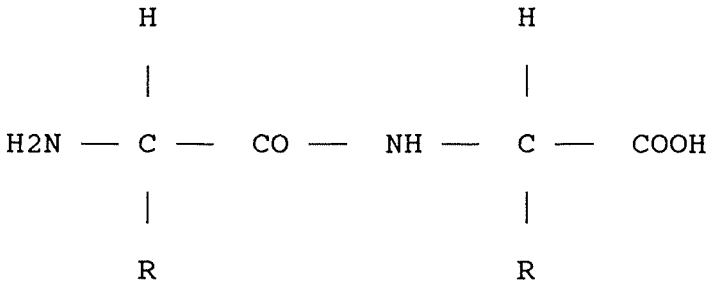


Figure 2 : Liaison entre deux acides aminés

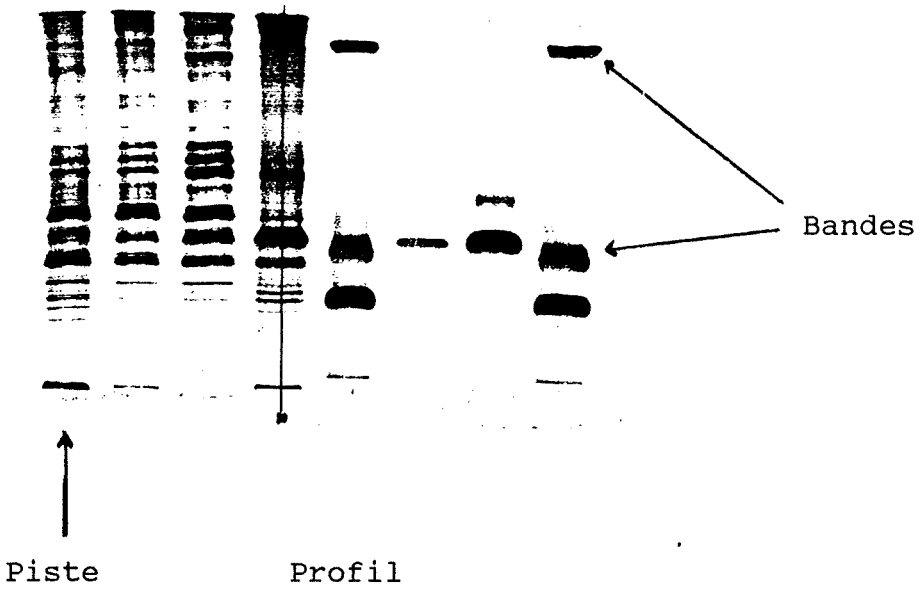


Figure 3 : Exemple de gel d'électrophorèse à une dimension.
(taille réelle)

3. LE TRAITEMENT D'IMAGES

Dans ce chapitre seront exposés les grands principes du traitement d'images. Les techniques utilisées dans ce travail seront vues de façon plus détaillée.

Le traitement de l'image d'un gel d'électrophorèse consiste à detecter et quantifier les bandes révélées par coloration. Ce processus correspond au problème de la reconnaissance des formes. Celui-ci est habituellement décomposé en trois étapes :

1. amélioration de l'image,
2. extraction des caractéristiques,
3. classification.

L'amélioration de l'image consistera dans notre cas essentiellement à l'élimination de détails inutiles, tels les bruits de fond, traînées, etc.... L'extraction des caractéristiques a pour but de détecter les formes présentes dans l'image. Ces caractéristiques seront utilisées dans le processus de classification.

En principe, le processus de digitalisation permet d'obtenir une image proche de l'image originale. Mis à part le bruit de haute fréquence, la dégradation de l'image due à la digitalisation est peu importante. Cependant, le gel présente souvent des défauts. Ce sont ces derniers que l'on s'efforcera de corriger par des techniques de traitement d'image.

Une image digitalisée se compose d'une série de points caractérisés par une valeur discrète de niveau de gris. Les techniques d'amélioration modifient les images, donnant à tout point une nouvelle valeur de niveau de gris.

Un premier groupe de techniques consiste à appliquer une transformation linéaire ou logarithmique à tout point de l'image originale. Ces techniques servent essentiellement à modifier les contrastes de l'image.

Un second type de traitement d'image consiste à modifier la répartition des niveaux de gris de l'image, c'est à dire agir sur son histogramme.

Aucun de ces deux types de traitement n'a été appliqué ici, les algorithmes de détection et de classification mis au point devant travailler sur des images aussi proches que possible de l'image originale, étant donné que l'on attend un résultat quantitatif. Par contre, les diverses techniques visant à éliminer le bruit de fond ont été appliquées, et plus particulièrement les techniques de lissage. Celles-ci recalculent le niveau de gris des points en fontion des valeurs de gris des points situés dans le voisinage. La principale fonction de ces filtres est d'éliminer la "neige", ou le bruit de haute fréquence.
Il est possible de mettre au point de nombreux filtres, dont l'utilité est fonction de la nature des défauts à corriger.
La recherche de solutions empiriques est inévitable.
Ci-dessous sont représentés quelques filtres classiques à deux dimensions. Les éléments des matrices filtres sont les coefficients par lesquels les valeurs des points auxquels le filtre est appliqué sont multipliées.

Dénomination	Coefficients	Effets
Moyenne directe	$\frac{1}{4} * \begin{matrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{matrix}$	Elimine les bruits de type neige Rend l'image trouble
	$\frac{1}{16} * \begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix}$	idem
Filtre médian	Le point central est remplacé par la valeur du point médian de la zone entourant le point	Elimine efficacement les valeurs aberrantes

Des variantes de ces filtres, ainsi que des algorithmes plus spécifiques seront explicités dans le chapitre concernant les principes de résolution.

4. LES FONCTIONNALITES DU SYSTEME

Le système d'analyse de gels d'électrophorèse à une dimension doit permettre de déterminer le poids moléculaire de constituants d'un mélange de molécules, ainsi que leur quantification.
Ceci se fait de la manière suivante :
L'échantillon à analyser est soumis à l'électrophorèse, sur la plaque de gel où est disposé également l'étalon, mélange de protéines de poids moléculaire connu et dont la quantité est connue, qui migreront si possible à équidistance l'une de l'autre, et sur toute la longueur de la piste, pour permettre d'établir de façon précise la distance de migration en fonction du poids moléculaire. La forme générale de la relation entre distance de migration et poids moléculaire est de la forme

$$DM = a + b * LOG(PM),$$

où DM est la distance de migration et PM le poids moléculaire. La piste étalon doit permettre de calculer les paramètres a et b de cette droite.
D'autre part, comme les protéines de l'étalon sont présentes en quantité connue, il est possible, en comparant les colorations des bandes, d'estimer la quantité de chaque constituant de l'échantillon.

Le système reçoit en entrée le gel ou l'image brute du gel à analyser. L'utilisateur doit pouvoir choisir la piste qui servira d'étalon, ainsi que le début et la fin de cette piste. Ceci afin de limiter les contraintes tant du côté du gel lui-même qu'au niveau de la prise de vue (standardisation). De même qu'il choisit lui-même le début et la fin de la piste à analyser. Le système fournit alors une liste reprenant les bandes détectées et les poids moléculaires estimés sur base de l'étalon.

5. CHOIX ET PRINCIPES DE RESOLUTION

5. 1. Introduction

D'emblée, il a fallu définir des voies de résolution du problème, comme par exemple décider de la nécessité ou non de débruiter l'entièreté de l'image ou de ne travailler que sur des profils qui sont des coupes longitidinales de pistes. D'autre part, comment utiliser les methodes mises au point pour les gels à deux dimensions ?
Dans ce chapitre seront d'abord expliquées les similitudes entre les deux types de gels, mais aussi leurs différences d'un point de vue traitement d'image essentiellement. Ensuite les principaux problèmes rencontrés et leurs solutions proposées seront explicitées.

5. 2. Comparaison entre les images de gels à une et à deux dimensions

Dans une électrophorèse à deux dimensions, nous avons vu que la séparation des constituants du mélange de protéines est très bonne, en tous cas bien meilleure que dans une électrophorèse à une dimension. Ceci entraîne une différence importante entre les images obtenues : Dans une électrophorèse à une dimension, les chevauchements de bandes seront beaucoup plus nombreux que les chevauchements de pics des images de gels à deux dimensions. Cependant, nous pouvons considérer que dans une même piste, les profils sont relativement semblables. La conséquence en est que l'information peut être extraite en ne considérant qu'un seul profil par piste, avec comme avantage substantiel un temps de

traitement sensiblement plus court. En ce qui concerne le bruit de fond, comme les traînées se forment parallèlement au sens de migration, le calcul du background peut se calculer sur un profil par piste.

Le choix de travailler piste par piste a été retenu, en utilisant un seul profil par piste. Pour rappel, une piste est un couloir de migration, le profil est la coupe longitudinale dans ce couloir, représentant ainsi en abcisses la distance de migration et en ordonnées l'intensité de la coloration, proportionnelle à la concentration en protéines. Les bandes sont l'équivalent des taches ou "spots" des gels à deux dimensions, et trahissent la position des protéines dans les pistes. En se basant sur des travaux précédemment effectués sur l'analyse de gels par traitement d'image, et après vérification sur quelques exemples, les profils des bandes, c'est-à-dire les coupes parallèles au sens de migration des protéines sont assimilés à des courbes de Gauss. Ceci est vrai évidemment tant que l'électrophorèse a été correctement effectuée et que les phénomènes de diffusion n'ont pas pris le pas sur les forces du champ électrique, les premières tendant à uniformiser la distribution des protéines dans le gel.

5. 3. Principaux problèmes à résoudre

5. 3. 1. L'amélioration de l'image

Deux processus très importants sont à effectuer avant l'étape d'extraction de l'information présente dans une piste :

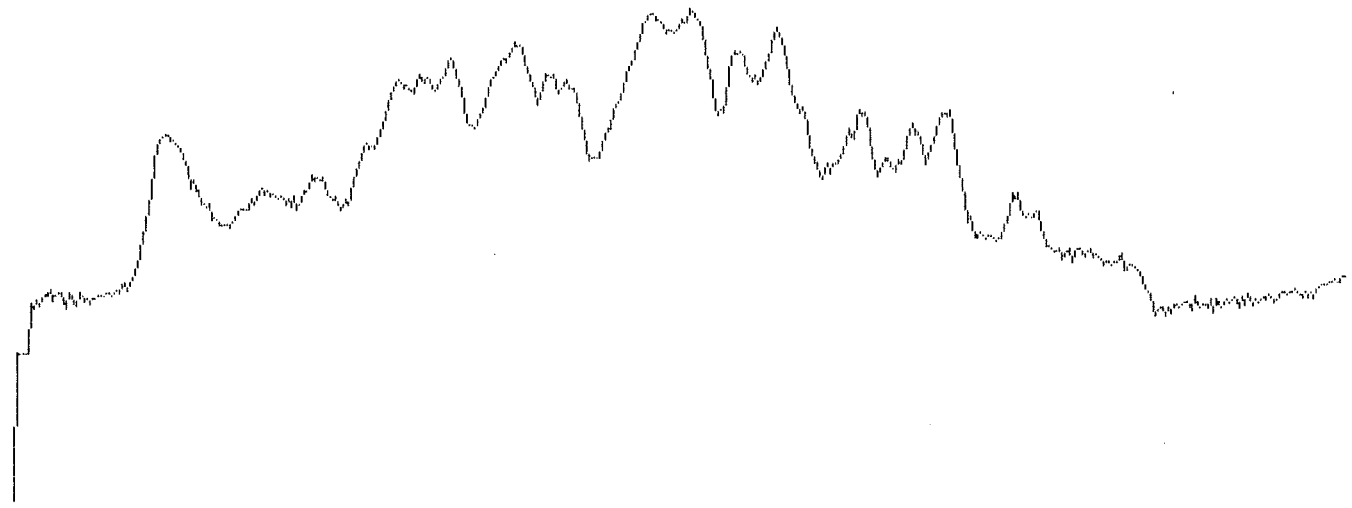
le lissage de l'image,
la soustraction du bruit de fond.

5. 3. 1. 1. Le lissage de l'image

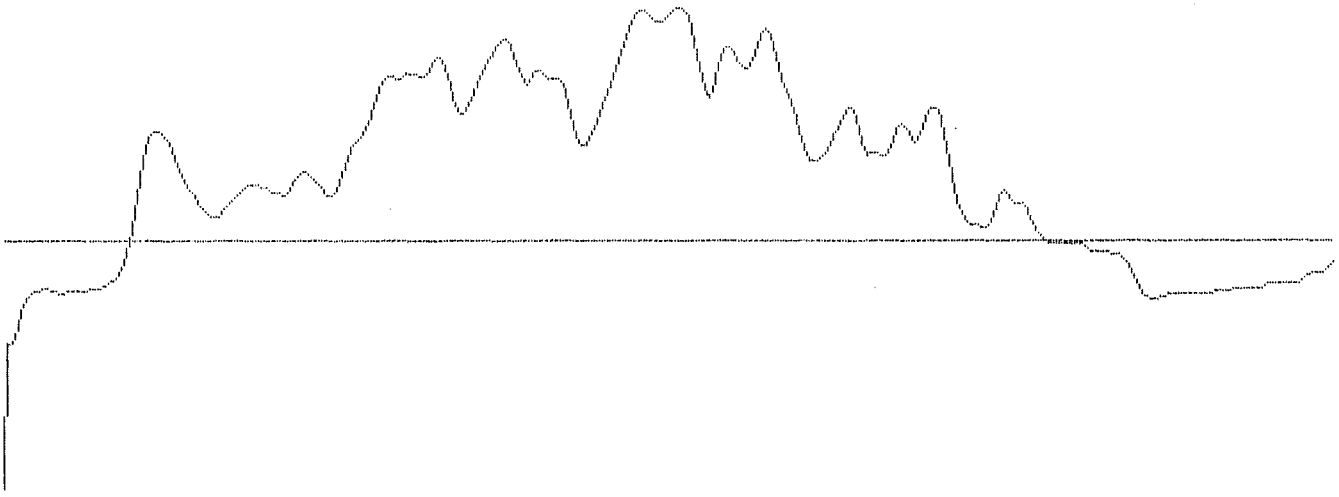
Pour le lissage de l'image, nous avons vu qu'il existe différents filtres, permettant de limiter l'impact des valeurs aberrantes. Des filtres "ligne" ont été mis au point. Pour le choix des filtres, il a fallu procéder à des essais sur des profils types. Un résultat acceptable est obtenu avec un filtre

1/4 * [1 2 1].

La figure suivante montre l'effet de ce filtre sur un profil.



a)



b)

Figure 4. Effet d'un filtre "121" sur un profil préalablement inversé. a : Image originale ; b : Image filtrée

5. 3. 1. 2. Le débruitage de l'image

Les bruits de fond de type neige peuvent être éliminés par des filtres. Il n'en est pas de même pour le bruit de fond véritable ou background. Celui-ci consiste en un fond plus ou moins sombre englobant toute l'image, pouvant montrer une variation lente et régulière d'un bout à l'autre de celle-ci.
De plus, il faut tenir compte des traînées. Ces deux types de bruits doivent être éliminés pour permettre d'obtenir les valeurs réelles des hauteurs des pics. En effet, la valeur de niveau de gris d'un point est la somme du background et sa valeur réelle. Il faut donc calculer le background en tout point du profil, ou de l'image, ensuite le soustraire. Cette opération est primordiale dans le cas de gels à deux dimensions, car dans ce dernier cas, il existe un gradient suivant les deux axes. Cependant, le calcul du background n'offre pas de dificulté particulière, étant donné que ces gels comportent de grandes zones dans lesquelles il n'y a pas de pics ; la valeur de gris de ces zones peut être considérée comme background. Dans le cas d'une électrophorèse à une dimension, il faut choisir un (ou éventuellement plusieurs) point dans chaque piste, dont l'intensité sera considérée comme niveau de background, car il faut déterminer le background piste par piste. Le calcul par des algorithmes du type de ceux utilisés pour l'électrophorèse à deux dimensions est plus aléatoire, du fait du chevauchement beaucoup plus important des bandes. Le principe de ce type d'algorithme est le suivant : chaque ligne ou colonne de l'image est parcourue, et la valeur du point lu est comparée à celle du point précédent. Tant que les deux valeurs sont

relativement semblables, elles sont considérées comme background ; lorsqu'apparaît une différence significative, celà traduit la présence d'un pic, et la dernière valeur avant l'entrée dans le pic est conservée. La sortie du pic se traduit par un retour de valeurs semblables et un retour à un niveau proche du niveau retenu avant l'entrée dans le pic. Le profil de la figure 4 laisse entrevoir les difficultés d'application d'un tel algorithme, au vu de l'importance du chevauchement des bandes.

5. 3. 2. L'extraction de l'information de l'image

5. 3. 2. 1. Introduction

L'image est maintenant à même de livrer les informations qu' elle contient. Nous savons qu'une piste d'électrophorèse est constituée de bandes. Du fait de leur chevauchement, il est très difficile, au jugé, de les discerner. Ceci représente un aspect du problème, l'autre étant la quantification. L'extraction de l'information se base sur une modélisation de l'image à analyser.

5. 3. 2. 2. Modélisation de l'image

Certains auteurs considèrent qu'une tache protéique dans un gel à deux dimensions ne peut être modélisée. L'estimation du "poids" d'un spot ou d'une bande dans un gel à une dimension nécessiterait de rechercher le contour, et de sommmer les intensités à l'intérieur de ce contour. Cette méthode est difficile à appliquer en cas de chevauchements de taches. Or, le chevauchement de bandes est loin d'être l'exception dans les gels à une dimension.
Plus généralement, il est admis que l'étalement d'une bande sous

l'effet des forces de diffusion est gaussien, la migration étant quant à elle responsable d'une certaine dissymétrie. Une coupe dans une bande suivant le sens de migration est donc une courbe constituée de deux demi-courbes de Gauss différentes lorque l'on travaille en intensité. Cependant, il n'est pas déraisonnable de modéliser un profil de bande par une seule courbe de Gauss.

5. 3. 2. 3. Détection du nombre de bandes et de leur position

L'hypothèse de la distribution gaussienne des protéines, donc de l'intensité de coloration des bandes de l'image peut être acceptée. Cependant, nous ne savons rien de la variance de ces distributions. Dans de nombreux cas, les courbes se chevauchent. Le profil de la piste représente la somme de toutes les courbes de Gauss individuelles représentant chaque bande. Certaines bandes seront bien visibles, parceque bien séparées des autres, d'autes en revanche, trop proches les unes des autres, ne se trahiront que par un épaulement sur le profil de piste. L'algorithme suivant permet de les détecter efficacement. Le problème revient à détecter les maxima locaux et les variations du sens de la concavité de la courbe du profil.
Le principe retenu est le suivant : le profil est parcouru, et les points sont lus trois à trois. soit X,Y,et Z trois points lus. Les valeurs du premier et du dernier point, soit X et Z sont comparées afin de déterminer le sens de la pente (non sa valeur). Le point central, soit Y, est utilisé pour déterminer le sens de la concavité de la courbe, en testant si sa valeur est supérieure ou inférieure à la moyenne des valeurs des points extrêmes. La figure suivante illustre cette manière de procéder.

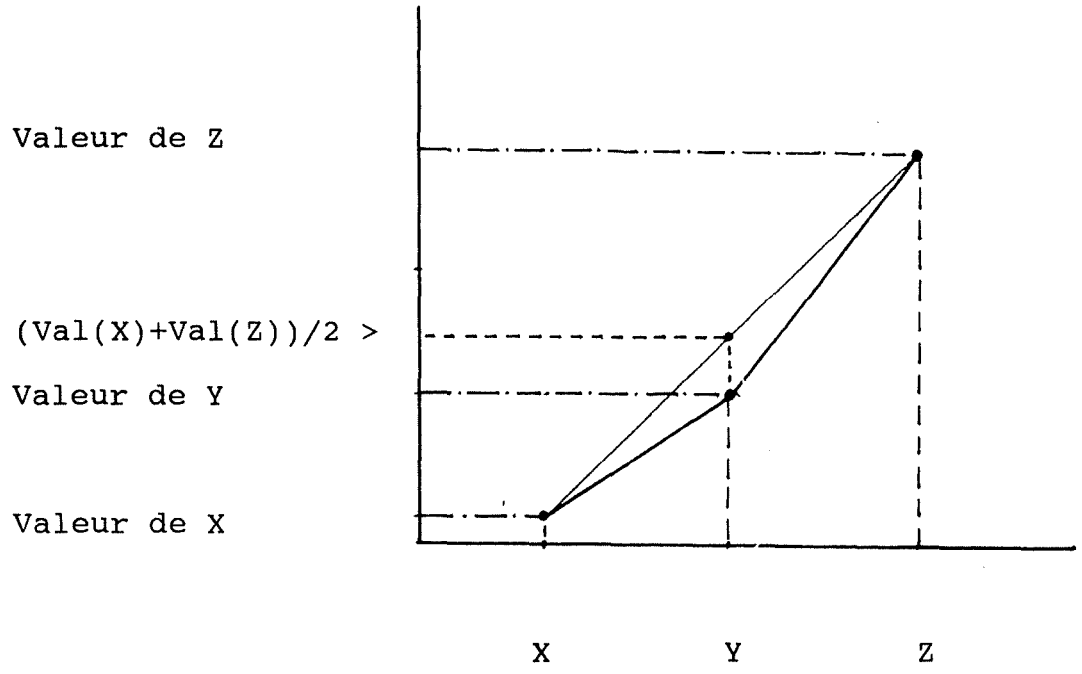


Figure 5. Principe de détermination du type de courbe

Quatre types de portions de courbe sont ainsi déterminées :

- courbe ascendante, concavité vers le haut, (3)
- courbe ascendante, concavité vers le bas, (1)
- courbe descendante, concavité vers le haut et (4)
- courbe descendante, concavité vers le bas. (2)

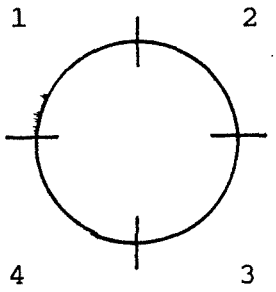
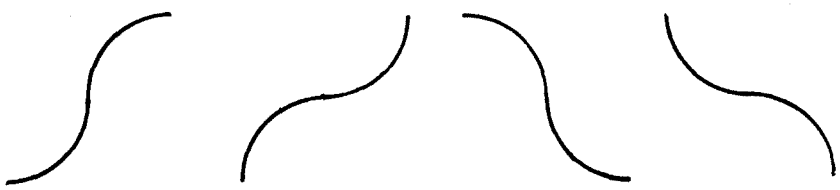


Figure 6. Types de courbes utilisés par l'algorithme détecteur de bandes

La combinaison de ces portions de courbes donne quatre types de points d'inflexion :

- sur pente ascendante, passage de concavité vers le haut à concavité vers le bas, (A)
- sur pente ascendante, passage de concavité vers le bas à concavité vers le haut, (B)
- sur pente descendante, passage de concavité vers le haut à concavité vers le bas, (C) et
- sur pente descendante, passage de concavité vers le bas à concavité vers le haut. (D)



A B C D

Figure 7. Les quatre types de points d'inflexion

La combinaison deux à deux de ces points d'infexion permet la détection des maxima locaux (A suivi de D) et des épaulements (A, B ou C, D consécutifs).

La figure suivante montre les sommets détectés par l'algorithme sur un profil de piste réel.

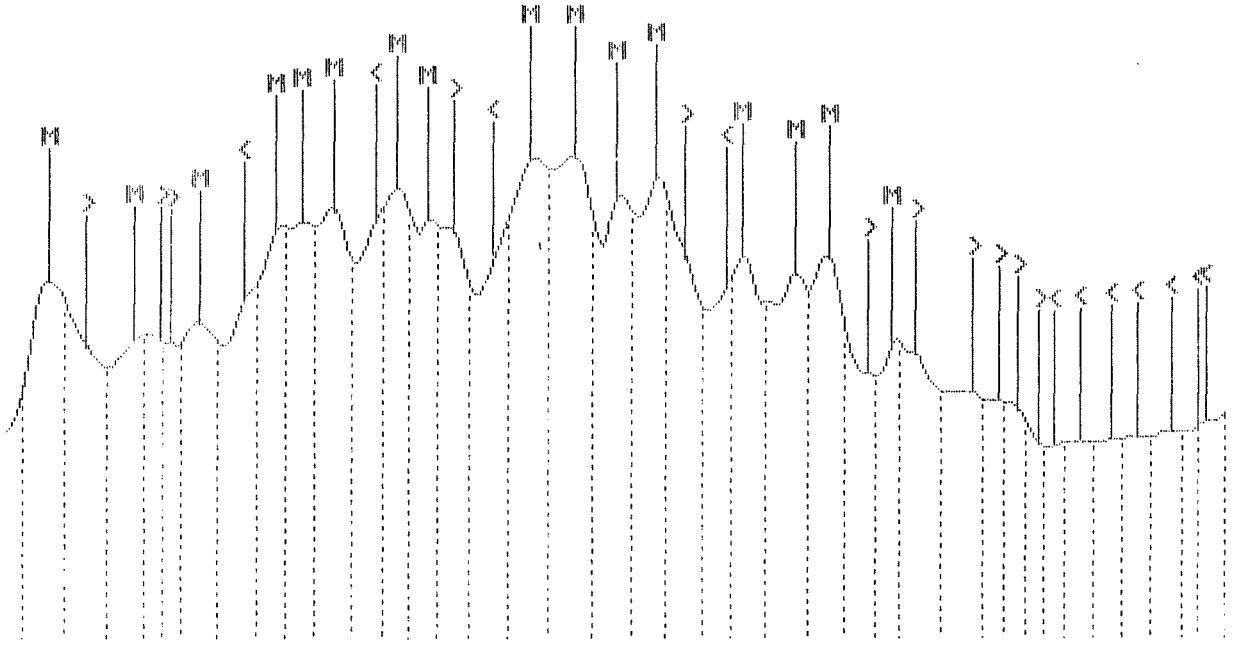


Figure 8. Sommets trouvés par l'algorithme de détection de bandes
M : maximum local, < : épaulement à gauche du maximum,
> : épaulement à droite du maximum local.

5. 3. 2. 4. Séparation des bandes et quantification

5. 3. 2. 4. 1. Introduction

Divers systèmes ont été développés pour séparer des spots ou taches en électrophorèse bidimensionnelle. Ces techniques sont adaptables à l'électrophorèse monodimensionnelle.

Un premier type de techniques se base sur la détection des contours ; nous en avons déjà parlé dans le paragraphe concernant la détection du nombre de bandes.

Un second type se base sur la modélisation des spots ou bandes par des courbes de Gauss ; c'est l'idée sur laquelle repose l'algorithme de séparation des bandes développé ici.

4. 3. 2. 4. 2. L'analyse de mélanges de distributions de même type

La séparation et la quantification des bandes revient, vu le modèle adopté, à analyser un mélange de distributions normales de probabilité. D'autres systèmes font appel au modèle de la courbe de Gauss pour quantifier les taches de protéines (V.Henin, P. Vincens). L'algorithme utilisé ici reprend le principe de l'algorithme développé par A. Schroeder (Schroeder 1975), utilisé dans le travail de V. Henin pour l'analyse de gels à deux dimensions. Cet algorithme analyse un mélange de distributions sur base d'un nombre de distributions fixé au préalable. Le principe de cet algorithme est le suivant :

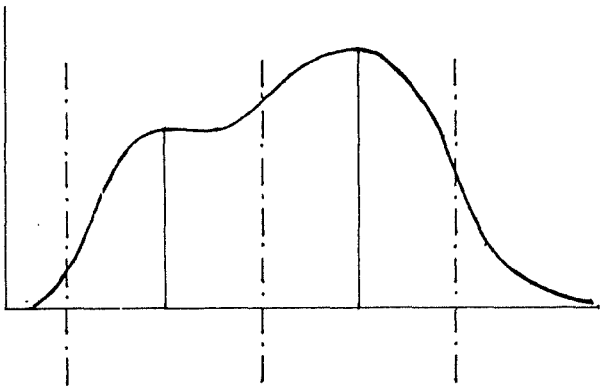
- Le nombre de distributions (de classes) et leur type est connu au préalable : ceci permet de prépartitionner l'échantillon sur base par exemple de la distance de chaque point par rapport au centre de chaque classe, pour autant que la position approximative de ce centre soit connue.
- Pour chaque classe de points, moyenne et variance sont estimées par la méthode du maximum de vraisemblance.
- Ensuite, les points sont partitionnés en classes sur base de leur distance associée à la variance de chaque distribution par rapport à la moyenne de chaque distribution, tant qu'un minimum local d'une fonction critère n'est pas atteint.

La convergence de cet algorithme est démontrée dans le rapport de recherhe de A. Schroeder (Schroeder, 1975).

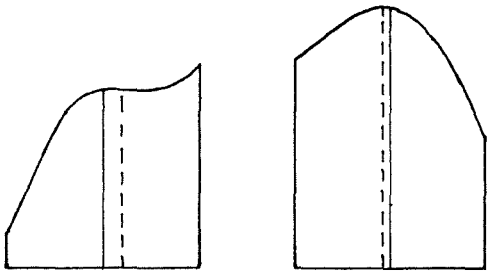
5. 3. 2. 4. 3. Algorithme de séparation des bandes

Si une tache protéique est modélisée par une courbe de Gauss, le profil obtenu peut être modélisé par une somme de courbes de Gauss. C'est sur cette idée que se base cet algorithme.
Tout comme dans l'application de l'algorithme de Schroeder à la résolution de gels à deux dimension de V. Henin, l'ensemble des points est d'abord prépartitionné. Ici, la prépartition est effectuée en prenant comme bornes des classe la mi-distance entre les positions des bandes préalablement détectées par la procédure de détection des centres. Ensuite, les distributions obtenues sont assimilées à des distributions normales dont moyenne et écart-type sont évalués. A partir de ces valeurs, chaque courbe de Gauss est construite, et les courbes obtenues sont sommées. Le profil

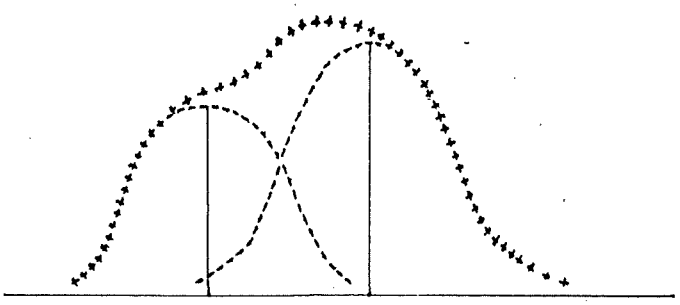
résultant est comparé au profil de départ. S'il s'en écarte significativement, les partitions sont reconstruites, d'après les distances au sens de la variance entre les points et chaque centre de partition. La procédure recommence tant que le nouveau profil résultant est meilleur.
Voici schématiquement la manière de procéder.



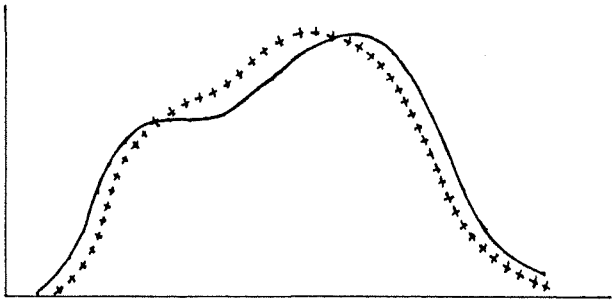
1) Calcul des bornes pour la prépartition en classes.



2) Calcul des paramètres des partitions : variances, nouveaux centres (moyennes de classe), poids (intégrales)



3) Détermination de courbes de Gauss (- - -), et somme de ces courbes (+ + +)



4) Comparaison entre le profil résultant (+ + +) et le profil de départ (---).

Sur des exemples concrets, l'algorithme converge rapidement, après trois ou quatre itérations, pour autant que le nombre de sommets soit raisonnable (pas plus de 50).

Voici les résultats sur le profil de la figure 4. :

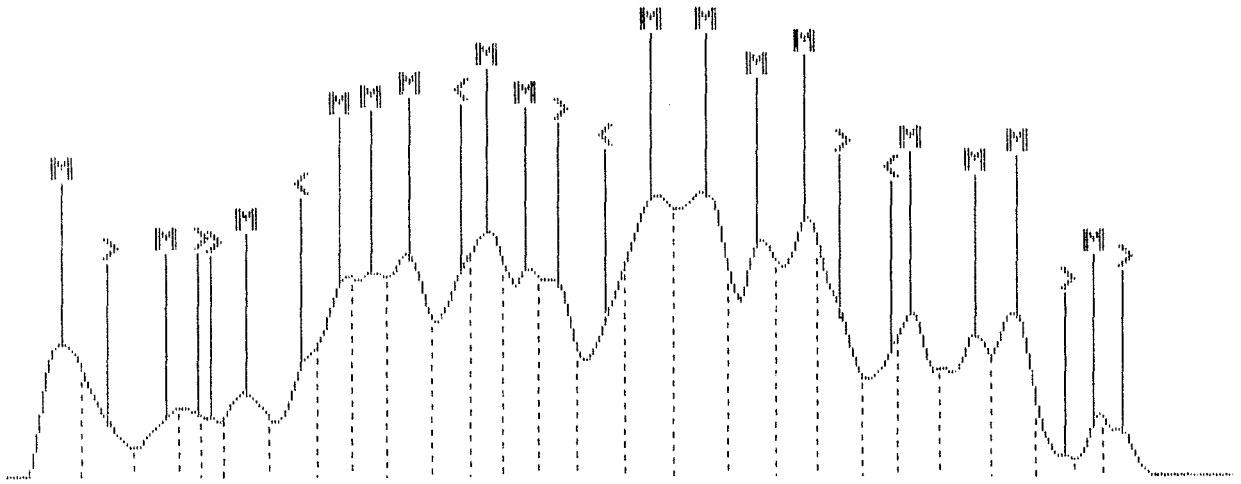


Figure 9. Prépartition des points en classes (traits pointillés)

RECONSTITUTION
Voici l'original

TERMINE

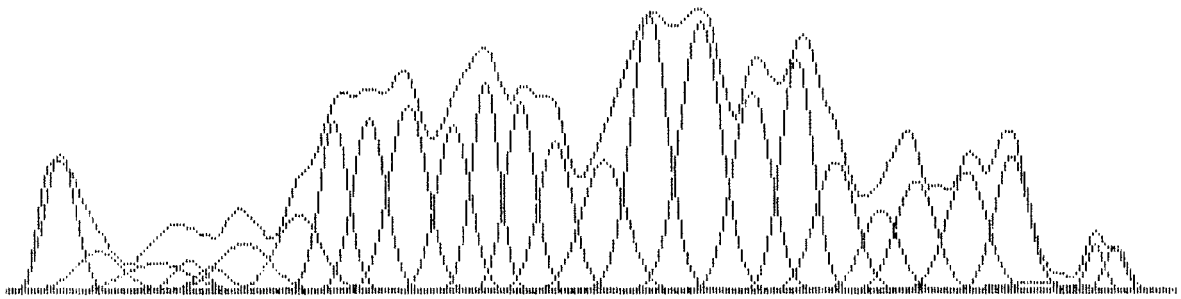


Figure 10. Courbes de Gauss calculées à la troisième itération, avec en superposition le profil original.

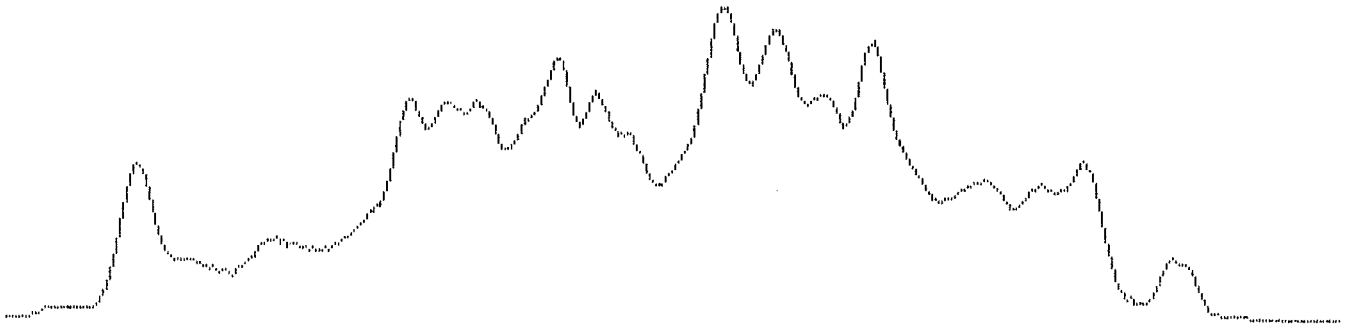


Figure 11. Profil résultant de la somme des courbes de Gauss

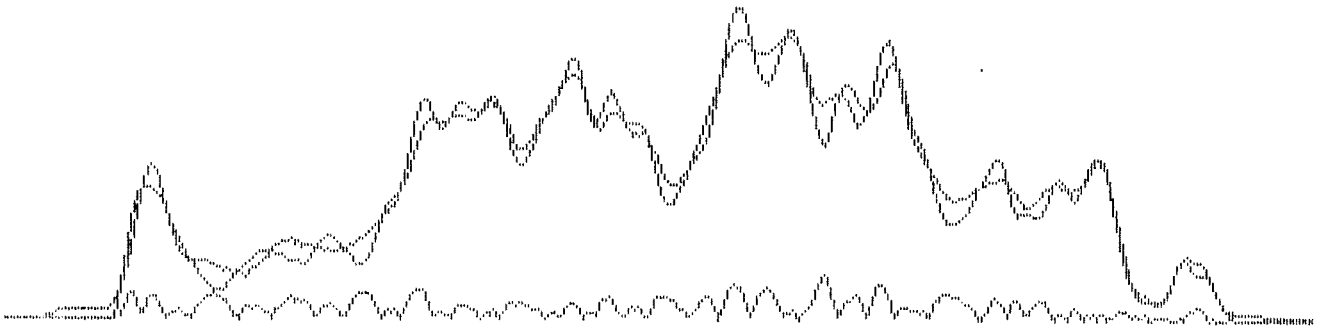


Figure 12. Profil résultant de la somme des courbes de Gauss trouvées, profil d'origine et différence

L'algorithme donne la position estimée et l'intégrale des courbes représentant une bande, ce qui permet une analyse quantitative des gels d'électrophorèse. IL reste à déterminer la fonction d'étalonnage.

5. 3. 3. L'étalonnage du gel.

Le principe de l'étalonnage d'un gel à une dimension a déjà été expliqué dans le chapitre 3 ; rappelons que le poids moléculaire est donné par la fonction

a + b (LOG DM), avec DM = distance de migration.

Il suffit de désigner, sur la piste étalon, les bandes qui révèlent des protéines connues, de PM connu et dont le dosage est connu. Ces données seront associées à la distance de migration, et une peréquation rectiligne pourra être effectuée.
Cependant, il reste quelques difficultés. Le gel n'est pas toujours rectangulaire. Les pistes peuvent être courbes, ou de longueurs différentes. Un remède serait d'utiliser plusieurs pistes étalon judicieusement disposées, mais cette solution a été rejetée afin de ne pas modifier le travail de fabrication du gel. La détection automatique des pistes n'a pas été retenue en raison de sa complexité. La solution retenue est de demander à l'utilisateur de désigner les limites des pistes. Ce choix implique que l'utilisateur soit familiarisé avec l'analyse de gels.

5. 3. 4. La structure des données manipulées.

Etant donné le choix de travailler sur des profils et non sur la totalité de l'image, il a été possible de travailler sur des tableaux de structures, par allocation de place mémoire. La structure de données de base est définie comme étant une zone, caractérisée par : début, fin, milieu, variance, poids....
Tant les profils que les courbes sont représentées par ces structures, représentées en annexe.

5. 4. Les limites du système

Que le gel à analyser soit un gel à une dimension ou à deux dimensions, il donne une image sur deux dimension. Il a été décidé de travailler sur des profils plutôt que sur l'image bidimensionelle. Le choix du bon profil est dès lors primordial, ce qui demande une attention toute particulière de l'utilisateur.
Dans sa forme actuelle, le système ne permet pas de tenir compte de la distorsion des pistes, ce qui entraîne inévitablement une erreur d'autant plus importante que le gel s'éloigne de la forme rectangulaire.
Les taches saturées, c'est à dire celles dont l'intensité est maximale sur de grandes distances ne peuvent évidemment être analysées correctement.
Enfin, le sytème ne tient pas compte d'une éventuelle inhomogénéité dans la vitesse de migration.
Ces conditions supposent un gel de bonne qualité, ainsi qu'une prise d'image acceptable.

6. LA STRUCTURE DU PROGRAMME

Les différentes procédures du programme peuvent se regrouper suivant leur type.

- Un premier groupe de fonctions s'occupe des échanges entre la mémoire vidéo, le mémoire centrale et les fichiers. Ces fonctions utilisent souvent des procédures de la bibliothèque fournie avec le matériel PCSCOPE. Les principales fonctionnalités offertes sont :

    - Digitalisation de l'image
    - Chargement d'une image à partir d'un fichier
    - Chargement d'un profil

- Un second groupe de fonctions permet d'améliorer un profil.

- Un troisième groupe de fonctions extrait l'information des profils

- Un quatrième goupe de fonctions transforme les informations extraites en informations utiles pour l'utilisateur ; il s'agit essentiellement des fonctions d'étalonnage et de calcul de poids moléculaire sur base de cette fonction.

- L'interface, enfin permet le dialogue avec l'utilisateur et la coordination des différentes groupes de fonctions. Un chapitre spécifique y est consacré.

Etant donné que l'application utilise des fonctions d'une bibliothèque externe, qui sont conçues pour fonctionner avec un type de matériel bien déterminé, il est difficile de la développer de manière à ce qu'elle soit indépendante du matériel utilisé. Pour pallier à cet inconvénient, les fonctions de la bibliothèque PCSCOPE n'apparaissent que dans le premier groupe de fonctions. Le travail de détection des bandes, leur séparation, les filtrages se font sur un profil, qui n'est autre qu'une liste de points. Dès le moment où le profil est lu en mémoire centrale, aucune action sur la mémoire image n'est strictement nécessaire. Cependant, en vue de faciliter le travail de l'utilisateur, des actions sur la mémoire image sont permises, notamment la détermination du bruit de fond et le positionnement d'un curseur en vue de désigner une bande particulière, ce qui est plus simple que sur un profil.

7. L'INTERFACE

Le rôle de l'interface est de permettre à l'utilisateur d'effectuer une ou plusieurs tâches à l'aide du système informatique mis à sa disposition. Le système informatique est un outil permettant l'optimisation du travail.
Le but de l'électrophorèse est la séparation de constituants d'un mélange. L'objectif de l'utilisateur d'un système d'analyse de gels d'électrophorèse est d'obtenir une interprétation de l'image obtenue. L'utilisateur sait comment interpréter une image de gel d'électrophorèse. Donc si le dialogue entre l'utilisateur et le système s'effectue en des termes communs, c'est à dire si à l'aide du système il peut procéder par les mêmes étapes que lors d'une interprétation visuelle, alors le système sera plus commode à utiliser. En effet, le problème central d'une interface réside dans la transformation des variables psychologiques en variables physiques et vice-versa. Moins il y a de différences entre ces deux types de représentation des connaissances, plus aisées seront ces transformations et plus facile seront l'utilisation et l'apprentissage de l'interface (Bodart 1991).
Ce système fait appel à deux écrans, l'écran de l'ordinateur et le moniteur vidéo externe. Les actions telles le choix de la piste étalon, la désignation des bandes dont on désire connaître la concentration et/ou le poids moléculaire se fait à l'aide de la souris, sur la mémoire image. Cette manière de procéder favorise l'implication de l'utilisateur et permet de se rapprocher d'un modèle d'interface construit selon la métaphore " du mini-monde ". L'interface en tant que tel tend à disparaître, les opérations

effectuées par l'utilisateur sont quasiment faites dans le domaine de la tâche.

Les champs d'entrée de données, menus, messages de service ou d'erreur apparaissent sur l'écran de l'ordinateur. L'instrument de dialogue est le clavier, la souris se réservant les manipulations sur la mémoire image.

Les messages informant l'utilisateur sur l'état du système seront relativement nombreux ; la durée des traitements, bien que réduite au maximum, risque en fait d'être suffisamment longue pour inquiéter l'utilisateur.

Au niveau du dialogue proprement dit, le système comporte un seul menu ; plus bas dans la hiérarchie des fonctionnalités, les choix sont souvent binaires (dichotomiques). Les champs d'entrée de données, messages de service ou d'erreur apparaissent dans des fenêtres distinctes.

Un aspect dont il faudra inévitablement tenir compte est l'ergonomie du poste de travail, à cause des deux écrans. Ceux-ci ne peuvent se trouver à trop grande distance l'un de l'autre.

8. DISCUSSION

8.1. Introduction

Le but de ce travail était la réalisation d'un système fiable et rapide d'analyse de gels d'électrophorèses à une dimension.
L'analyse de gels d'électrophorèse à une dimension est utilisée de façon routinière, et la technique d'électrophorèse bidimensionnelle ne connaît pas un développement tel qu'il avait été prédit il y a quelques années en raison notamment de la complexité de l'analyse des résultats, ainsi que de la difficulté de mise au point des outils d'aide à l'interprétation des données.
Le système développé ici permet une analyse plus précise que l'analyse traditionnelle de gels à une dimension, tout en utilisant um matériel relativement simple.
Le volume mémoire nécessaire est faible, le système ne manipulant que des coupes longitudinales de pistes, c'est-à-dire des tableaux de 512 éléments au maximum si on utilise une caméra de résolution 512 points sur 512. La puissance de calcul nécessitée tant par les traitements d'amélioration de l'image que par les algorithmes d'extraction de l'information est également faible. Ceci a pour conséquence que le traitement de gels de taille beaucoup plus importante (les gels utilisés dans le cadre de ce travail ont une taille d'environ 50 cm2) peut être envisagé sans crainte. Pour des gels de taille plus importante, il est en effet nécessaire de disposer d'une plus grande résolution spatiale. Cet aspects différencient fondamentalement ce système de ceux qui ont été développés pour les gels d'électrophorèse à deux dimensions.

8.2. Performances de l'algorithme de quantification des bandes

L'algorithme de quantification des bandes est un algorithme itératif dont le nombre d'itérations dépend du nombre de bandes détectées par la procédure de détection des bandes. chaque itération contient les sous-procédures de prépartition, de paramétrage, de partition et de calcul de courbe résultante à partir des courbes de Gauss individuelles.
Les complexités théoriques sont de l'ordre de O(n), où n représente le nombre de points constituant un profil. Ces complexités théoriques déterminent la complexité théorique globale (O(n)) de l'algorithme de quantification des bandes. Rappelons que le nombre de points dans un profil est ici de 512 au maximum.

8.3. L'organisation des données

La possibilité de sauvegarder une image bidimensionnelle a été abandonnée en raison de la place mémoire nécessaire. Seules les informations données par l'utilisateur concernant le gel, les profils et les résultats sont sauvegardés. Les informations utilisateur telles la date, le nom de l'expérimentateur... et les résultats (poids moléculaires estimés, intensités relatives) sont sauvés dans un fichier d'extension .RES, les profils dans un fichier d'extension .GEL (un byte par point). La place mémoire nécessitée est de ce fait limitée.

8.4. Perspectives

Actuellement, le système n'est pas à même de traiter des gels de mauvaise qualité (saturation, distorsion des pistes, bruit de fond excessif), mais moyennant quelques adaptations, la possibité peut être envisagée d'assouplir quelque peu les conditions requises pour le gel. Dans ce cas, l'automatisation de l'analyse paraît être l'étape suivante, d'autant plus que l'expérience dans le développement de systèmes intelligents d'aide à l'interprétation des données pour les gels à deux dimensions s'accumule, et pourrait pourrait s'avérer utile.

BIBLIOGRAPHIE

Appel, R., Funk, M.

MELANIE

Un système expert d'aide à l'analyse et à l'interprétation d'électrophorèses bidimensionnelles.

Press. Univ. Nancy.

Bodart, F.

Cours introductif aux interfaces homme-machine.

Facultés Universitaires N.D. de la Paix, Namur,

Année académique 1990-1991.

Garrels, J. I.

Two dimensionnal gel electrophoresis and computer analysis of proteins synthesized by clonal cell lines.

J. Biol. Chem. 254 (16), 7961-7077.

Henin, V.

Analyse quantitative de gels d'électrophrèse par traitement d'image.

Mémoire, Facultés Universitaires N.D. de la Paix, Namur

Institut d'informatique, Année académique 1988-1989.

Scapin, L. D.

Guide ergonomique de conception des interfaces homme-ordinateur.

I.N.R.I.A., Le Chesnay.

Schroeder, A.
Analyse d'un mélange de distributions de probabilités de même type.
I.R.I.A., Rapport de recherche n° 104, 1975.

Vincens, P.
Analyse informatisée des gels d'électrophorèse bidimensionnelle.
E.N.S. Laboratoire de Zoologie, Paris, 1987.

Youg, T. Y., Coraluppi, G.
Stochastic estimation of a mixture of normal density functions using an informatic criterion.
I.E.E.E., Transactions on Information Theory Vol. IT-16, n°3, 1970.

ANNEXES

Table des matières Page

A1. Description des procédures du système

Afficheligne (unsigned char *l, unsigned int n)

type : procédure graphique ;

paramètres : adresse de la zone mémoire contenant les valeurs (niveaux de gris) d'une ligne, nombre de valeurs ;

description : affiche à l'écran le tracé d'un profil.

Affichesommets(Z *z)

type : entrée-sortie ;

paramètres : adresse d'une structure "zone" décrivant une ligne ;

description : affiche les caractéristiques de la ligne.

Affichesommetsg(Z *z)

type : procédure graphique ;

paramètres : adresse d'une structure "zone" décrivant une ligne;

description : affiche, sur le tracé du profil, les sommets(centres de bandes) et leur zone d'infuence.

Background(Z *z)

type : traitement de profil ;

paramètres : adresse d'une structure "zone" décrivant une ligne;

description : sur base d'une valeur de bruit de fond établie au préalable, corrige le profil.

Camera()

type : gestion entrées-sorties;

paramètres : /

description : initialise la mémoire vidéo, prépare le hardware pour l'acquisition d'image et permet la saisie de l'image.

Chargeligne(unsigned char *fichier, **p, int nb, long debut)

type : gestion entrées-sorties ;

paramètres : fichier contenant les profils,
adresse de zone mémoire à réserver pour y stocker le profil,
nombre de points du profil(longueur) et
position du premier point de ce profil ;

description : chargement en mémoire centrale d'un profil de ligne.

Choisir piste()

type : interface ;

paramètres : /

description : permet à l'utilisteur de choisir une piste (la colonne et le début) sur l'écran vidéo à l'aide de la souris.

Choisir fin piste()

type : interface ;

paramètres : /

description : permet à l'utilisateur, après avoir choisi une piste et son début, de choisir sa fin à l'aide de la souris.

Copie(int colonne,int depart, int longueur)

type : gestion des entrées-sorties ;

paramètres : ordonnée de piste,
abcisse du début de la piste,
longueur de la piste en pixels ;

description : sauve le profil extrait de la piste sur base des indications fournies par l'utilisateur dans un fichier.

Creeligne(K *s, int n)

type : traitement de profil ;

paramètres : adresse d'une structure décrivant une courbe, longueur de la courbe ;

description : crée une courbe de Gauss sur base d'une moyenne, variance et un nombre d'observations.

Detectsommet(Z *z)

type : traitement de profil ;

paramètres : adresse de la structure "zone" décrivant une ligne ;

description : détecte les sommets de bandes sur base des variations d'inflexion du profil.

Erreur(FILE *f, int n)

type : gestion entrées-sorties ;

paramètres : fichier recherché, numero d'erreur ;

description : production d'un message d'erreur en cas de problème de lecture du fichier.

Etalonner()

type : interface ;

paramètres : /

description : effectue l'étalonnage du gel sur base des données fournies par l'utilisateur.

Filtre1(unsigned char *l, unsigned int n)

type : traitement de profil ;

paramètres : adresse d'un profil en mémoire centrale, longueur de ce profil ;

description : lisse le profil en recalculant chaque point en fonction de ses deux voisins de gauche et deux voisins de droite selon un masque "1-2-2-2-1".

Filtre2(unsigned char *l, unsigned int n)

type : traitement de profil ;

paramètres : adresse d'un profil en mémoire centrale, longueur de ce profil ;

description : lisse le profil en recalculant chaque point en fonction de ses voisins selon un masque "1-2-1".

Inverse(unsigned cha *l, unsigned int n)

type : traitement d'un profil ;

paramètres : adresse d'un profil en mémoire centrale, longueur de ce profil ;

description : inverse les niveaux de gris du profil.

Lire pistes()

type : interface ;

paramètres : /

description : effectue le transfert de la mémoire vidéo à la mémoire centrale des lignes choisies par l'utilisateur, et les sauve dans un fichier.

Lire te()

type : interface ;

paramètres : /

description : permet à l'utilisateur de sélectionner les bandes étalon à l'aide de la souris, et d'entrer le poids moléculaire correspondant au clavier.

Parametrage(Z *z)

type : traitement de profil ;

paramètres : adresse de la structure "zone" contenant un profil ;

description : effectue une première partition des points du profil et calcule ensuite la moyenne (position réelle du centre), variance et poids (nombre total d'observations dans une distribution réelle, ici somme des valeurs de gris des points). Sur base de ces premières estimations, produit les courbes de Gauss modélisant chaque partition.

Reduction(Z *z)

type : traitement de profil ;

paramètres : adresse de la structure "zone" contenant un profil ;

description : additionne les courbes de Gauss modélisant chaque partition et compare le résultat obtenu au profil initial; cette procédure boucle tant que l'écart entre la courbe modèle et le profil d'origine diminue. Après la réalisation de la condition d'arrêt, les résultats sont affichés à l'écran.

Reduit(Z *z)

type : traitement de profil ;

paramètres : adresse de la structure "zone" contenant un profil ;

description : recalcule les partitions de points en fonction des distances à la moyenne, en fonction des variances et poids calculés, et construit les courbes de Gauss correspondantes.

Remplir(mp)

type : interface ;

paramètres : pointeur sur le menu appelant ;

description : permet l'entrée de données de service.

Resultat(Z *z)

type : procedure graphique ;

paramètre : adresse de la structure "zone" contenant un profil ;

description : affiche à l'écran

- le profil original filtré,
- les courbes de Gauss calculées à la dernière itération de la procédure "reduction",
- la somme de ces courbes (profil résultant),
- la différence entre les deux profils.

A2. Structures manipulées

Les structures utilisées par les procédures de traitement de profil sont définies comme suit :

- une structure qui recevra les diverses formes du profil appelée **Z** dont les champs sont
  - 1) un pointeur sur le profil d'origine ;
  - 2) un pointeur sur le profil obtenu après modélisation ;
  - 3) un pointeur sur le profil résultant de la différence entre les deux premiers ;
  - 4) la longueur du profil ;
  - 5) le nombre de sommets de bande, c'est à dire le nombre de courbes de Gauss qui modélisent ce profil ;
  - 6) un pointeur sur ces courbes, et
  - 7) les résultats.
- des structures destinées à contenir les courbes de Gauss modélisant chaque sommet de bande :
  - 1) un pointeur sur la ligne origine ;
  - 2) la limite gauche de la partition modélisée ;
  - 3) la limite droite de cette partition ;
  - 4) sa largeur ;
  - 5) son centre ;
  - 6) sa variance ;
  - 7) son écart-type ;
  - 8) son poids (intégrale) ;
  - 9) le type de sommet (épaulement gauche, maximum local, épaulement droit sur le profil origine).

A3. L'utilisation du logiciel

A3.1. Hard- et software nécessaires au fonctionnement du Logiciel

Afin de pouvoir utiliser le programme d'analyse de gels à une dimension GEL1D il est nécessaire de disposer :

* d'un PC ou XT fonctionnant sous MS-DOS version 3.10 ou supérieure
* d'une carte image
* d'une caméra de résolution 512 sur 512 points
* d'un moniteur video de résolution égale ou supérieure à 512 points sur 512
* d'une souris.

Ce logiciel est conçu pour fonctionner avec le matériel d'acquisition et de traitement d'images PCSCOPE fourni par IDS.

A3.2. Chargement du programme

S'assurer que tous les périphériques sont correctement raccordés et allumés.

Il est vivement conseillé d'installer le programme sur un disque dur. Les instructions qui suivent supposent que le programme se trouve dans le répertoire de base du disque dur (c:\).

Taper **rtscope gel1d** puis appuyer sur **Enter**.

A3.3. Le menu principal

Le menu (figure A1) apparaît alors à l'écran. A l'aide des touches curseur, choisir l'option **PRISE DIRECTE** ou **IMAGE SUR DISQUE** suivant que l'on désire analyser un gel ou une image stockée sur disque.

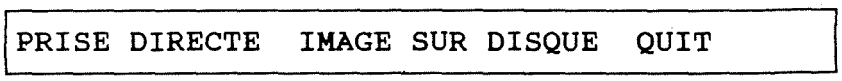


Figure A1. Menu principal

A3.3.1. L'option PRISE DIRECTE

A3.3.1.1. Entrée des renseignements généraux

Une fenêtre (figure A2) apparaît alors sur la partie gauche de l'écran. Le champ "Identificateur" doit être complété ; Il est impossible de quitter ce champ tant qu'il n'a pas reçu au moins un caractère. L'identificateur sera le nom sous lequel les différents fichiers seront sauvegardés.

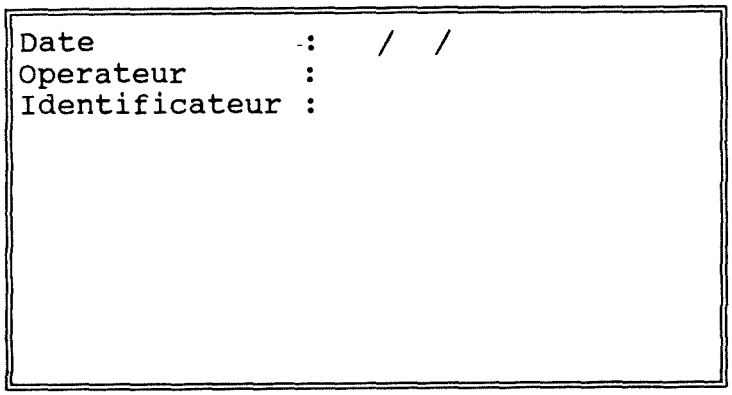


Figure A2. Fenêtre d'entrée de données d'ordre général

A3.3.1.2. La prise de vue

Après avoir complété la fenêtre des renseignements généraux, l'enfoncement d'une touche quelconque ouvre une fenêtre sur la partie droite de l'écran (figure A3). La carte vidéo est initialisée et la caméra entre en fonctionnement. A ce moment, il faut placer le gel par rapport à la caméra de façon à ce que que l'image occupe la plus grande surface possible à l'écran et qu'elle soit rectangulaire. L'enfoncement d'une touche fige l'image, indiquant de ce fait qu'elle est acquise par le système.

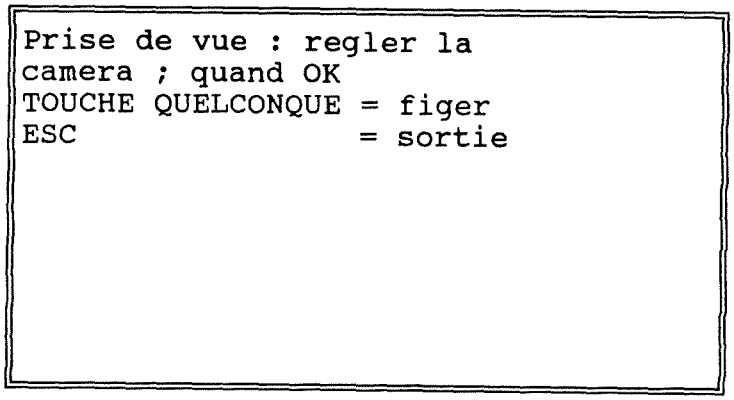


Figure A3. Fenêtre de dialogue avec l'utilisateur

A3.3.1.3. Le choix de la piste étalon et des bandes étalon

Les actions sur l'image se font à l'aide de la souris ; un curseur sur l'écran vidéo visualise la position de la souris. L'enfoncement du bouton droit valide le choix de la position. Choisir d'abord le début de la piste étalon. Le choix de la position du début est primordial, il détermine la position de la coupe longitudinale dans le profil qui sera utilisée pour l'analyse. Il faut veiller à ce qu'elle correponde autant que possible à l'axe de la piste. Choisir ensuite la fin de la piste. Le processus de sélection des bandes est indentique, mais à chaque bande sélectionnée doit correspondre une valeur de poids moléculaire à entrer à l'aide du clavier.
Les coefficients de la droite d'étalonnage sont alors affichés (alpha et beta).

A3.3.1.4. Choix des bandes à analyser

Les choix des bandes à analyser se fait de la même manière que le choix des bandes étalon, mais il faut au préalable donner le nombre de bandes que l'on désire analyser. La marche à suivre est indiquée dans la fenêtre, étape par étape.

A3.3.2. L'option IMAGE SUR DISQUE

Cette option permet de travailler sur l'image d'un gel, stockée sur disque. Les étapes à suivre sont identiques que pour l'option PRISE DIRECTE, sauf bien sûr en ce qui concerne le chargement de l'image dans la mémoire vidéo. Il suffit de donner le répertoire et le nom du fichier à lire.

Remarque : le sauvetage d'images de gels sur disque est déconseillé en raison du volume mémoire nécesaire.

A3.3.3. Quitter le programme

L'option QUIT du menu permet l'abandon du programme. En cours de programme, l'enfoncement de la touche ESCAPE provoque l'abandon des traitements courants et le retour au menu.

A3.4. Affichage et impression des résultats

Les résultats sont affichés à l'écran par séries de 20 bandes si le nombre de bandes est supérieur à 20; les résultats s'affichent comme suit :

Numéro de la bande	PM estimé	Quantité relative par rapport à la piste

et peuvent être imprimés.

Code source du programme

```
typedef unsigned char C;
typedef unsigned int I;
typedef unsigned long L;
typedef double D;

typedef struct courbe { C *ligne ;
                        I gauche ;
                        I droite ;
                        I largeur ;
                        I centre ;
                        D variance ;
                        D ecart;
                        L poids ;
                        C tipe;    /* </M/> */
                      } K ;

typedef struct zone { C *original ;
                      C *obtenu ;
                      C *soust ;
                      I largeur ;
                      L poids ;
                      C nbcourbes ;
                      K *courbes ;
                      I resultat ;
                    } Z ;

struct tabsommets { I position;
                    L integrale;
                    L PM;
                    float qte_rel;
                  } ;

struct tab_etalon { I dm;
                    D lpm;
                  } ;


struct tabsommets ts[100];
struct tab_etalon te[6];




#define PI 3.14159265359

#include <vcstdio.h>
#include <malloc.h>
#include <math.h>
#include <memory.h>
#include <float.h>

VCMENU *menunew(), *menuxnew();
```

```
/* * * * * * * * * *  Variables fenetres * * * * * * * * * */

int   wdo_remplir;
COUNT Panneau2;
COUNT Panneau3;


/* * * * * * * * * *  Autres variables * * * * * * * * * * */

TEXT fichsave[9];
TEXT fichretr[29];
TEXT courant[9];
TEXT Date_jour[10+1];
TEXT Operat[18+1];
TEXT Id_gel[8+1];
TEXT lutinverse[256];
COUNT valdate();
COUNT Prise_directe();
COUNT lire_image();
COUNT Quitter();
COUNT isblank();
COUNT num_bim;
COUNT num_lut;
COUNT num_hdr;
int   debut;
int   fin;
int   numpiste;
int   longueur;
D     alpha,beta;
COUNT nombrepiste;


VCMENU *PanPt;

D R,pi2;




void stat(void);
extern char analyser(char *fichier,I nblignes,I debut,C *p);
```

```
main()
    {
    vcstart(CLRSCRN);
    set_colors();
    prep_menus();
    empty(Date_jour,10+1);
    empty(Operat,18+1);
    numpiste=0;
    nombrepiste=0;

    debut=0;
    fin=0;
    num_bim=-1;
    empty(fichsave,9);
    empty(fichretr,29);
    empty(courant,9);
    empty(Id_gel,8+1);
    vcmenu(PanPt);
    vcend(CLOSE);

    }

/*****************************************************************/
/**************** Perequation ********************************/

void stat()
{
  double sumxi, sumxisq, sqsumxi,sumyi,sumxiyi;
  I i;
  double ata[2][2];
  double aty[2];
  double detata;
  double invata[2][2];
  sumxi=0;
  for(i=0;i<6;i++)
   { sumxi=sumxi+te[i].lpm;}
  atsaynum(2,0,&sumxi,"########",REAL);
  getone();
  sumxisq=0;
  for(i=0;i<6;i++)
   { sumxisq=sumxisq+(te[i].lpm*te[i].lpm);}
  sqsumxi=sumxi*sumxi;
  atsaynum(2,0,&sqsumxi,"########",REAL);
  getone();
  ata[0][0]=6;
  ata[0][1]=sumxi;
  ata[1][0]=sumxi;
  ata[1][1]=sumxisq;
  detata=6*sumxisq-sqsumxi;
```

```
  if (detata!=0)
   {
   invata[0][0]=ata[1][1]/detata;
   invata[0][1]=(-ata[0][1])/detata;
   invata[1][0]=(-ata[1][0])/detata;
   invata[1][1]=ata[0][0]/detata;
   }
  else atsay(15,0,"Impossible : det nul ");
  sumyi=0;
  for(i=0;i<6;i++)
   { sumyi=sumyi+te[i].dm;}
  sumxiyi=0;
  for(i=0;i<6;i++)
   { sumxiyi=sumxiyi+(te[i].lpm*te[i].dm);}
  aty[0]=sumyi;
  aty[1]=sumxiyi;

  alpha = invata[0][0]*aty[0]+invata[0][1]*aty[1];
  beta = invata[1][0]*aty[0]+invata[1][1]*aty[1];
 }

/***************** Entree donnees ***************************/

void remplir()
   {
   wdo_remplir=wxxopen(0,0,12,38,"",
                       ACTIVE|BORDER|BD1|CURSOR,
                       11,37,0,32);
   if( wdo_remplir == -1 )
      terror("Erreur systeme. Programme interrompu");
   empty(fichsave,9);
   empty(courant,9);
   xatsay(0,0,"Date            :",vc.dflt);
   xatsay(1,0,"Operateur       :",vc.dflt);
   xatsay(2,0,"Identificateur :",vc.dflt);
   xxatgetc(vcdgt, wdo_remplir, 0, 17, Date_jour, "99/99/9999",
                   valdate, "Date du jour", "", vc.dflt, vc.dflt,
                   DEFAULT, STRING, NULL, NULLFUNC, NULLFUNC);
   xxatgetc(vcdgt, wdo_remplir, 1, 17, Operat, "Xxxxxxxxxxxxxxxxxx",
                   NULLFUNC, "Blanco", "", vc.dflt, vc.dflt,
                   DEFAULT, STRING, NULL, NULLFUNC, NULLFUNC);
   xxatgetc(vcdgt, wdo_remplir, 2, 17, Id_gel, "XXXXXXXX",
                   isblank, "", "", vc.dflt, vc.dflt,
                   DEFAULT, STRING, NULL, NULLFUNC, NULLFUNC);
   readgets();
   strcpy(fichsave,Id_gel);
   strcat(fichsave,".WDO");
   wtodisk(wdo_remplir,fichsave);
   strcpy(courant,Id_gel);
   empty(fichsave,9);
   atsay(10,0,"touche qcq pr continuer");
   getone();
   wclose(wdo_remplir);

   }
```

```
void choisir_fin_piste()
 {
   short bstat,x2,y2,xdeb,ydeb,colstat,ligstat;
   char libelle[2];
   xdeb=100; ydeb=100;

    xatsay(9,0,"choisissez la fin de la piste ",vc.dflt);
    delay(1);
    xdeb=100; ydeb=100;
    M_GETPOS(&bstat,&x2,&y2,xdeb,ydeb,0,10,-1,libelle);
    fin=y2;
    werase(Panneau2);
 }

void choisir_piste()
 {
   short bstat,x2,y2,xdeb,ydeb,colstat,ligstat;
   char libelle[2];
   xdeb=100; ydeb=100;

   xatsay(8,0,"choisissez le debut de la piste ",vc.dflt);

   M_GETPOS(&bstat,&x2,&y2,xdeb,ydeb,0,10,-1,libelle);

   numpiste=x2; debut=y2;
   choisir_fin_piste();
 }

/***************  Remplir le tableau d'etalonnage ****/

void lire_te()
{
   unsigned int cpt;
   short bstat,x2,y2,xdeb,ydeb,colstat,ligstat;
   char libelle[2];
   D poids_mol,controle;
   xdeb=100; ydeb=100;
   werase(Panneau2);
   xatsay(0,0,"CHOISISSEZ LES BANDES ETALON     ",vc.dflt);
   xatsay(1,0,"A L'AIDE DE LA SOURIS ET ENTREZ  ",vc.dflt);
   xatsay(2,0,"LE PM CORREPONDANT AU CLAVIER    ",vc.dflt);
   getone();
   werase(Panneau2);
```

```
   for(cpt=0;cpt<6;cpt++)
   {
    poids_mol=0.0;
    xatsay(1,0,"choisissez une bande",vc.dflt);
    M_GETPOS(&bstat,&x2,&y2,xdeb,ydeb,0,10,-1,libelle);
    te[cpt].dm=y2;
    xatsay(2,0,"                    ",vc.dflt);
    xatsay(3+cpt,0,"Donnez son PM ",vc.dflt);
    atgetc(3+cpt,15,&poids_mol,"##########",DEFAULT,REAL|FLDBLANK,NULL);
    readgets();
    te[cpt].lpm=poids_mol;
    controle=2*poids_mol;
    atsaynum(2,0,&controle,"########",REAL);
   }
   stat();
 werase(Panneau2);

}

void camera()
{
  initfgb();
  presetfgb(200);
  videoeur();
  selnumcam(3);
  syncext();
  initfgb();
  syncext();
  cacquire();
  inkeyb();
  frmfreez();
}

void etalonner()

{
  int longueur_et;
  choisir_piste();
  longueur_et=abs(fin-debut)+1;
  atsay(6,0,"longueur ");
  atsaynum(6,10,&longueur_et,"###",INTEGER);
  atsay(7,0,"debut ");
  atsaynum(7,10,&debut,"###",INTEGER);
  atsay(8,0,"fin ");
  atsaynum(8,10,&fin,"###",INTEGER);
  delay(3);
  werase(Panneau2);
  lire_te();
  atsay(11,0, "Alpha et Beta : ");
  atsaynum(11,17,&alpha,"######",REAL);
  atsaynum(12,17,&beta, "######",REAL);
  getone();
  werase(Panneau2);
}
```

```
/********************** pour pistes non etalon **************/


void lire_pistes()
{
  int i;
  unsigned char p[600];
  short bstat,x2,y2,xdeb,ydeb,colstat,ligstat;
  char libelle[2];
  FILE *fp,*fopen();
  atsay(0,0,"CHOISISSEZ LA PISTE A ANALYSER ET");
  atsay(1,0,"SA LONGUEUR A L'AIDE DE LA SOURIS");
  atsay(2,0,"BARRE D'ESPACEMENT = COMMENCER   ");
  atsay(3,0,"UTILISEZ LE BOUTON GAUCHE DE LA  ");
  atsay(4,0,"POUR SELECTIONNER : 1) LE DEBUT, ");
  atsay(5,0,"                    2) LA FIN    ");
  getone();
  werase(Panneau2);
  i=1;
  xdeb=100; ydeb=100;
  atsay(0,0,"DONNEZ LE NBRE DE PISTES ");
  atgetc(0,30,&nombrepiste,"9", DEFAULT, INTEGER, NULL);
  readgets();
  strcat(courant,".GEL");
  fp=fopen(courant,"w");
  while(i<=nombrepiste)
        {
         werase(Panneau2 );
         choisir_piste();
         longueur=abs(fin-debut)+1;
         atsay(6,0,"longueur ");
         atsaynum(6,10,&longueur,"###",INTEGER);
         atsay(7,0,"debut ");
         atsaynum(7,10,&debut,"###",INTEGER);
         atsay(8,0,"fin ");
         atsaynum(8,10,&fin,"###",INTEGER);
         xinhyinc();
         getlncar((char far *)&(p[0]),numpiste,debut,longueur);
         fwrite(p,longueur-1,1,fp);
         i++;
        }
  fclose(fp);
}
```

```
prep_menus()
{
   addvcmstyle("STDHORIZ",HORIZONTAL|TITLECENTER,80,
               vc.cyan+vc.bold+(vc.bg*vc.blue),
               vc.blue+(vc.bg*vc.white),
               vc.cyan+(vc.bg*vc.blue),
               vc.white+(vc.bg*vc.blue),
               0 );
   addvcmstyle("STDVERT",VERTICAL|TITLECENTER,-1,
               vc.cyan+vc.bold+(vc.bg*vc.blue),
               vc.blue+(vc.bg*vc.white),
               vc.cyan+(vc.bg*vc.blue),

               vc.white+(vc.bg*vc.blue),
               0 );
   PanPt=menuxnew(22,0,80,"",
                        HORIZONTAL|TITLECENTER,
                        vc.cyan+vc.bold+(vc.bg*vc.blue),
                        vc.blue+(vc.bg*vc.white),
                        vc.cyan+(vc.bg*vc.blue),
                        vc.white+(vc.bg*vc.blue),
                        0,"NO HELP");
   menuxitem(PanPt,"PRISE DIRECTE",NULL,0,Prise_directe,NULL,"","",
                                                                  STRPA
   menuxitem(PanPt,"IMAGE SUR DISQUE",NULL,0,lire_image,NULL,"","",STRPA
   menuxitem(PanPt,"QUIT",NULL,0,Quitter,NULL,"Quitter","",STRPARM);
}


Panneau2_func()
{
   remplir();
   Panneau2=wxxopen(0,41,12,79,"",
                    ACTIVE|BORDER|BD1|CURSOR,
                    11,37,0,32);
   if( Panneau2 == -1 )
      terror("Probleme de logiciel : interruption");
   xatsay(0,0,"Prise de vue : regler la  ",vc.dflt);
   xatsay(1,0,"camera ; quand OK         ",vc.dflt);
   xatsay(2,0,"TOUCHE QUELCONQUE = figer ",vc.dflt);
   xatsay(3,0,"et acquerir l'image       ",vc.dflt);
   xatsay(3,0,"ESC                = sortie",vc.dflt);
   getone();
   camera();
   werase(Panneau2);
   xatsay(4,0,courant,vc.dflt);
   etalonner();
   lire_pistes();
   getone();
   analyser(courant,longueur,debut,0);

}
```

```
Prise_directe(mp)
VCMENU *mp;
   {
   Panneau2_func();
   return(0);
   }


lire_image(mp)
VCMENU *mp;
   {

   Panneau2=wxxopen(0,41,12,79,"",
                    ACTIVE¦BORDER¦BD1¦CURSOR,
                    11,37,0,32);
   if( Panneau2 == -1 )
      terror("Erreur fatale, programme interrompu");
   return(0);
   }


Quitter(mp)
VCMENU *mp;
   {
   vcend(CLOSE);
   }
```

```
#include <stdio.h>
#include <float.h>
#include <math.h>
#include <stdlib.h>
#include <dos.h>
#include <stdarg.h>
#include <graph.h>

typedef unsigned char C;
typedef unsigned int I;
typedef unsigned long L;
typedef double D;

typedef struct courbe { C *ligne ;
                        I gauche ;
                        I droite ;
                        I largeur ;
                        I centre ;
                        D variance ;
                        D ecart;
                        L poids ;
                        C tipe;    /* </M/> */
                      } K ;

typedef struct zone { C *original ;
                      C *obtenu ;
                      C *soust ;
                      I largeur ;
                      L poids ;
                      C nbcourbes ;
                      K *courbes ;
                      I resultat ;
                    } Z ;

#define PI 3.14159265359
D R,pi2;



/*************************************************************************
struct videoconfig vc;



/*****************  ACCES DISQUE   ************************************


void erreur(FILE *f,int n)
{ int e;
  e=errno ;

  printf("\nERREUR DE LECTURE %d" ,e) ;
  getch() ; fclose(f) ;
}
```

```
char chargeligne(C *fichier,C **p,int nb,L debut)
{ FILE *f;
  *p=(C *)malloc(nb);
  if (!*p) {
             printf("\npas a.c. de memoire");
             getch(); return(1);
           }
  f=fopen(fichier,"r+b");
  if (!f)
          {
             printf("\nFICHIER PAS TROUVE") ;
             fclose(f) ;
             return(1) ;
          }
  fseek(f,debut,SEEK_SET);
  if (fread(*p,nb,1,f)==0)
          { erreur(f,0) ; return(1) ; }
  fclose(f); return(0);
}


/*****************************************************************/
/*       GESTION GRAPHIQUE EN GENERAL                            */
/*****************************************************************/


/*************  AFFICHAGE DE LIGNES A L'ECRAN ******************/

void affiche ligne(C *l,I n)
{ I c,a=0,b=199,i=0;
  while (i++<n)
     {
        c=199-((I) (* (l++) )/2) ;
        _moveto(a,b);
        _lineto(i,c) ;
        a=i ; b=c ;
     }
}



/**************  AFFICHAGE DES SOMMETS SUR LE PROFIL **********/

void affichesommetsg(Z *z)
{ I ns,x,y;
  C *l,t[2];
  K *s;
  t[0]=32; t[1]=0;
  ns=z->nbcourbes;
  s=z->courbes;
  l=z->original;
  _setvideomode(_HRESBW) ;
  affiche ligne(z->original,z->largeur);
```

```
  printf("Il y a %d sommets",ns);
  while (ns--)
      {
        x=(I)(s->centre) ;
        t[0]=s->tipe ;
        _setlinestyle(0xFFFF) ;
        y=199-((I)(*(l+x))-50) ;
        _moveto(x,y+(50));
        _lineto(x,y) ;
        printf(t) ;
        _setlinestyle(0xAAAA) ;
        x=(I)((s++)->droite) ;
        _moveto(x,199-( (*(l+x))));
        _lineto(x,199) ;
      }
  getch();
  _clearscreen(_GCLEARSCREEN);
  _setvideomode(_DEFAULTMODE);
}


/*****************  CARACTERISTIQUES DES PICS *****************/

void affichesommets(Z *z)
{ char i=3;
  char n=0;
  K *s;
  I x,ns;
  C *p;
  s=z->courbes ;
  ns=z->nbcourbes;
  printf("\n\nIl y a %d sommets de poids total %ld\n",ns,z->poids);
  printf("\n\nFrappez une touche "); getch();
  while (ns--)
    {
        n++;
        printf("\n\n%d : [%u %u %u] poids : %ld",n,
                s->gauche,s->centre,s->droite,s->poids) ;
        printf("\n   variance %lf",s->variance) ;
        printf("\tecart %lf",s->ecart) ;
        printf("\n");
        x=s->largeur ; p=z->original+s->gauche ;
        while (x--) printf("%d ",*p++);
        s++ ; i-- ;
        /* if (!i) { getch() ; i=3 ; printf("\n"); } */
    }
  printf("\n\nFIN - frappez une touche") ; getch();
}




/***************** TRACE D'UNE COURBE DE GAUSS ***************/
```

```
void creeligne(K *s,I l)
{
  C *p;
  I m,n=1 ;
  D d1,d2 ;
  L nn,y;
  p=s->ligne ;
  m=l ;
  while (m--)  *p++=0 ;
  if (!s->ecart) return ;
  d1=pi2*s->ecart ;
  d2=(D)2*s->variance ;
  m=s->centre ;
  p=s->ligne+m ;
  y=(L)(R+(D)s->poids*exp(0)/d1) ;
  if (y>255) *p=255 ;
  else *p=(C)y ;
  while ((y>=1) && (n<l))
    {
      nn=n*n ; y=(L)(R+(D)s->poids*exp(-(D)nn/d2)/d1) ;
      if (y>255) y=255 ;
      if (m+n<l) *(p+n)=(C)y ;
      if (((signed int)m-(signed int)n)>=0) *(p-n)=(C)y;
      n++ ;
    }
}


/*****************    FILTRAGE    ************************************/


void filtre1(C *l,I n)
{ n-=4 ;
  while (n--) *(l+2)=((((I)(*l))<<1)+(((I)(*(l+1)))<<2)+
                      (((I)(*(l+2)))<<2)+(((I)(*(l+3)))<<2)+
                      (((I)(*((l++)+4)))<<1))>>4 ;
}

void filtre2(C *l,I n)
{
  n-=2 ;
  while (n--) *(l+1)=(((I)(*l))+(((I)(*(l+1)))<<1)+((I)(*((l++)+2))))>>2
}

void inverse(C *l,I n)
{
  while (n--)
     {
       *l=(255-(*(l)));
       l++;
     }
}

void bckgnd(C *l,I n)
```

```
{
  while (n--)
       {
         if (*l >30) (*l)=((I)(*l)-30);
         else
             (*l)=0;
         l++;
       }
}


/*****************  TRAITEMENTS    ******************************/


char detectsommet(Z *z)
{ C *r,*rr,t,*l,p0=5,p1;
  C i0,i1='A';
  I m0,m1,m2,cc=0,n1=0,n0=0,n;
  signed int w,m,d0=0,d1;
  K *ss;
  n=z->largeur ;
  l=z->original ;
  rr=r=(C *)malloc(n*3) ;
  if (!r) return(1);
  p0=0 ;
  cc=0 ;
  i0='B' ;
  while (n--)
    {
      m1= (I)(*l) + (I)(*(l+1)) ;
      m2= (I)(*(l+3)) + (I)(*(l+4)) ;
      m= m1+m2 - ( ((I)(*((l++)+2))) << 2) ;
      p1=!!m ;
      p1= p1 + ((p1&&(m1>m2))<<1) + ((p1&&(m<0))<<2) ;
      i1=0 ;
      if (p1 != p0)
         switch (p0)
                     { case 1 : { if (p1==5)  i1='B' ; break ; }
                       case 3 : { if (p1==7)  i1='C' ; break ; }
                       case 5 : { if (p1==1)  i1='A' ; break ; }
                       case 7 : { if (p1==3)  i1='D' ; break ; }
                     }
      if (p1) p0=p1 ;
              t=0 ;
      switch (i1)
            {
              case 0 : break ;
              case 'A' : {
                           if (i0=='B') t='<' ;
                           else
                             { n1=n0 ; i0=i1 ; }
                           break ;
                         }
              case 'D' : {
```

```
                        if (i0=='B') t='M' ;
                        else if (i0=='C') t='>' ;
                             else { n1=n0 ; i0=i1 ; }
                        break ;
                      }
              default :  { n1=n0 ; i0=i1 ; }
            }
       if (t) {
                m=n1+((n0-n1)>>1) ;
                *r++=(C)m ;
                *r++=(C)(m>>8) ;
                *r++=t ;
                cc++ ;
                n1=n0 ;
                i0=i1 ;
              }
     n0++;
   }
  if (!cc)
    {
      cc++ ;
      m=n1+((n0-n1)>>1) ;
      *r++=(C)m ;
      *r++=(C)(m>>8) ;
      *r='M';
    }
  ss=z->courbes=(K *)malloc(cc*sizeof(K)) ;
  z->nbcourbes=cc ;
  r=rr ;
  n0=0 ;
  if (!ss)
    { free(rr) ; return(1) ; }
  while (cc--)
    {
      ss->centre=n=*((I *)r)+2 ;
      r+=2 ;
      ss->tipe=*r++ ;
      ss->gauche=n0 ;
      if (cc) ss->droite=n1=((*(I *)r-n)>>1)+n ;
      else ss->droite=n1=z->largeur-1 ;
      ss->largeur = n1+1-n0 ;
      n0=n1+1 ;
      ss->ligne=(C *)malloc(z->largeur+6) ;
      if (!(ss++)->ligne) return(1) ;
    }
  free(rr) ; return(0) ;
}
```

```
/******************** CALCUL DES PARAMETRES DE DISTRIBUTION ***/
```

```
void parametrage(Z *z)
 {
   D v,t,tt=0 ;
   I larg,d,n,c,ns,x,m;
   C *p,*pl,*po,*obt;
   K *s;
   L moy ;
   ns=z->nbcourbes ;
   s=z->courbes ;
   larg=z->largeur ;
   obt=z->obtenu ;
   m=larg ;
   po=obt ;
   pl=z->soust ;
   while (m--)
     { *po++=*pl++=0 ; }
   while (ns--)
     {
       n=s->largeur ;
       p=z->original+(d=s->gauche) ;
       v=0;
       t=0;
       moy=0 ;
       while (n--)
         {
           moy+=x=(I)*p++ ;
           t+=(D)x*(D)(d++) ;
         }
       tt+=s->poids=moy ;
       if (s->poids)
         {
           n=s->largeur ;
           p=z->original+(d=s->gauche) ;
           s->centre=c=(I)(0.5+t/((D)moy)) ;
           while (n--)
             {
               t=(D)(d++)-(D)c ;
               v+=(D)(*(p++))*t*t ;
             }
           s->variance=v/=(D)moy ;
           s->ecart=sqrt(v) ;
         }
       else
         {
           s->centre=s->gauche ; s->variance=s->ecart=0 ;
         }
      creeligne(s,larg) ;
      m=larg ;
      pl=(s++)->ligne ;
      po=obt ;
      while (m--)
         if ((I)(*po)+(I)(*pl)<255) (*po++)+=(*pl++) ;

         else
```

```
            { (*po++)=255 ; pl++ ; }
      }
   z->poids=tt ;
   m=larg ;
   p=z->original ;
   pl=z->soust ;
   while (m--) *(pl++)=(C)abs((signed int)(*p++)-(signed int)(*obt++)) ;
}

/******************** AFFICHE RESULTATS GRAHIQUES ************/

I resultat(Z *z)
{
   K *s;
   I ns,larg;
   C *p;
   _setvideomode(_HRESBW);
   printf("RECONSTITUTION");
   printf("Voici l'original");
   larg=z->largeur ;
   afficheligne(z->original,larg) ;
   getch();
   printf("Sommets ");
   ns=z->nbcourbes ;
   s=z->courbes ;
   while (ns--) afficheligne((s++)->ligne,larg) ;
   printf("TERMINE") ;
   getch() ;
   _clearscreen(_GCLEARSCREEN);
   printf("Somme des sommets");
   afficheligne(z->obtenu,larg) ;
   getch();
   printf(" Superposition et soustraction");
   afficheligne(z->original,larg) ;
   afficheligne(z->soust,larg) ;
   printf(" Erreurs : %d/%ld",z->resultat,z->poids);
   getch() ;
   _clearscreen(_GCLEARSCREEN);
   _setvideomode(_DEFAULTMODE);
   return(ns);
}

I reduit(Z *z)
{
   K *s,*ss;
   C *obt,*or,*sous,*p,*po,*pl;
   D point,v,t,moy ;
   L poids ;
   I cx,cy,cc,d,larg,n,c,ns,x,m,nb;
   L res;
   n=larg=z->largeur ;
   nb=z->nbcourbes ;
   ss=z->courbes ;
   or=z->original+larg-1 ;
   sous=z->soust ;
```

```
obt=z->obtenu+larg-1 ;
printf("\n\nAnalyse des colonnes");
while (n--)
  {
  ns=nb ; s=ss ;
  if ((*obt) && (*or))
     {
       point=((D)*or)/((D)*obt) ;
       while (ns--)
          { p=(s++)->ligne+n ;
                      if (*p) { d=(I)(((D)(*p))*point) ;
                                if (d>255) *p=255 ;
                                else (*p)=d ;
                              }
                      else (*p)=0 ;
          }
     }
  else while (ns--) *((s++)->ligne+n)=0 ;
  *(obt--)=0 ; or-- ;
}
ns=nb ;
s=ss ;
printf("\nAnalyse des sommets : ");
while (ns--)
  {
    n=larg ;
    v=moy=0.0 ;
    poids=0 ;
    p=s->ligne+n-1 ;
    printf("%d ",ns);
    while (n--) if (d=*p--)
        { poids+=(L)d ; moy+=(D)d*(D)n ; }
    n=larg ; p=s->ligne+larg-1 ;
    if (poids)
        { s->poids=poids ;
          s->centre=c=(I)(0.5+moy/((D)poids)) ;
          while (n--) if (d=*p--) {
                                   cc=abs(c-n) ;
                                   v+=((D)cc)*((D)cc)*(D)d ;
                                  }
          s->variance=v/=(D)poids ;
          s->ecart=sqrt(v) ;
          n=larg ;
          creeligne(s,larg) ;
          p=s->ligne+larg-1 ;
          obt=z->obtenu+larg-1 ;
          while (n--) if ((I)(*obt)+(I)(*p)<255) (*obt--)+=(*p--) ;
                      else { (*obt--)=255 ; p-- ; }
        }
    else
       {
           s->poids=s->centre=s->variance=s->ecart=0.0 ;
           while (n--) *p--=0;
       }
    s++ ;
```

```
    }
  n=larg ;
  or=z->original;
  sous=z->soust;
  obt=z->obtenu ;
  res=0 ;
  while (n--)
      {
         c=(C)abs((signed int)(*or++)-(signed int)(*obt++)) ;
         *(sous++)=(C)c ; res+=c ;
      }
  return(res);
}

void reduction(Z *z)
{ I larg,nb,ns,n=1,t0=99999999 ;
  I i=0,t1=99999998;
  K *s,*sm,*ssm,*sz;
  C *obt,*p,*sous;
  _setvideomode(_TEXTBW80);
  printf("\nReduction en cours\n") ;
  nb=z->nbcourbes ;
  larg=z->largeur ;
  sz=z->courbes ;
  ssm=(K *)malloc(sizeof(K)*nb) ;
  while (t1<t0)
    {
      ns=nb ; s=sz ; sm=ssm ;
      while (ns--)
        {
          sm->centre=s->centre ;
          sm->poids=s->poids ;
          sm->variance=s->variance ;
          (sm++)->ecart=(s++)->ecart ;
        }
     t0=t1 ; t1=reduit(z) ;
     printf("\nL'iteration %d donne une erreur de %d",n++,t1);
    }

  printf("\n\nTERMINE  Voici les donnees") ;
  z->resultat=t0 ;
  n=larg ;
  obt=z->obtenu ;
  while (n--) *obt++=0 ;
  ns=nb ;
  s=sz ;
  sm=ssm ;
  printf("\n\n Il y a %d sommets",nb);
  while (ns)
    { i=23 ;
      while (ns && i)
       {
         printf("\nSOMMET %3d",nb-ns);
         s->centre=sm->centre ;
         s->poids=sm->poids ;
```

```
            s->variance=sm->variance ;
            s->ecart=sm->ecart ;
            printf("  centre %3d",s->centre);
            printf("  poids total %5ld",s->poids);
            printf("  variance    %lf",s->variance);
            creeligne(s,larg) ;
            if (s->ecart)
                {
                   n=larg ;
                   p=s->ligne ;
                   obt=z->obtenu ;
                   while (n--) if ((I)(*obt)+(I)(*p)<255) (*obt++)+=(*p++)
                               else { (*obt++)=255 ; p++ ; }
                }
            sm++; s++ ; i-- ; ns-- ;
      }
      getch();
   }
   n=larg ;
   sous=z->soust ;
   p=z->original ;
   obt=z->obtenu ;
   while (n--) *(sous++)=(C)abs((signed int)(*p++)-(signed int)(*obt++))

   resultat(z) ;

   free(ssm) ;
}

/*****************  PROGRAMME PRINCIPAL   ******************************

char traiteligne(char *fichier, I nblignes, I debut, C *p)
{
   Z z;
   K *s;
   I n,a;
   _setvideomode(_HRESBW);
   z.soust=(C *)malloc(nblignes);
   z.obtenu=(C *)malloc(nblignes);
   if ((!z.soust) || (!z.obtenu)) return(1) ;
   z.largeur=nblignes-6;
   printf("Chargement et filtrage ... ");
   if (!p)
      {
         z.original=(C *)malloc(nblignes);
         if (!z.original) return(1) ;
         if (chargeligne(fichier,&z.original,nblignes,debut))
            return(1);
      }
   else  z.original=p ;
   afficheligne(z.original,z.largeur);
   printf("\nligne originale");
   getch();
   _clearscreen(_GCLEARSCREEN);
   inverse(z.original,z.largeur) ;
```

```
  afficheligne(z.original,z.largeur);
  printf("\nligne inversee");
  getch();
  _clearscreen(_GCLEARSCREEN);
  filtre1(z.original,z.largeur) ;
  afficheligne(z.original,z.largeur);
  printf("\nfiltre 121");
  getch();
  _clearscreen(_GCLEARSCREEN);
  filtre2(z.original,z.largeur) ;
  afficheligne(z.original,z.largeur);
  printf("\n Filtre double");
  getch();
  _clearscreen(_GCLEARSCREEN);
  bckgnd(z.original,z.largeur) ;
  afficheligne(z.original,z.largeur);
  printf("\nImage debruitee");
  getch();
  _clearscreen(_GCLEARSCREEN);
  if (detectsommet(&z)) return(1);
  affichesommetsg(&z); getch();
  printf("\nParametrage en cours");
  parametrage(&z) ;
  /*  affichesommets(&z);  */
  reduction(&z) ;
  n=z.nbcourbes ;
  s=z.courbes ;
  while (n--)
  free((s++)->ligne) ;
  free(z.soust) ;
  free(z.obtenu) ;
  free(z.courbes) ;
  if (!p) free(z.original) ;
  return(0) ;
}

void analyser()
{ C *pp,*p;
  D x,y,z,*t;
  I n,m;
  K s;
  pi2=sqrt((D)2*(D)PI);
  R=0.9999999 ;
  if(_setvideomode(_HRESBW)==0)
     {printf("\nerreur de driver");exit(0);}

  traiteligne("lignes",512,0,0);
  traiteligne("lignes",512,512,0);
  traiteligne("lignes",512,1024,0);
  traiteligne("lignes",512,1536,0);
  traiteligne("lignes",512,2048,0)  ;

  _setvideomode (_DEFAULTMODE);
}
```